IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

MARQUETTE	Examiner: Vu, Viet Duy
Appl. Serial No. 09/965,057	Art Unit: 2514
Filed: September 26, 2001	Confirmation No.: 2669
For: MEDIA SESSION FRAMEWORK USIGN A CONTROL MODULE TO DIRECT AND MANAGE APPLICATION AND SERVICE SERVERS	

APPEAL BRIEF

Mail Stop: APPEAL BRIEFS - PATENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief supports the appeal from the decision of the Examiner mailed on April 11, 2003, finally rejecting claims 1-24 of the above-identified patent application.

Real Party in Interest

SandCherry Networks, Inc., as assignee of the above-identified patent application, (Reel/Frame 012213/0623), having an address of 1715 38th Street, Boulder, CO 80301, is the real party in interest.

Related Appeals and Interferences

Neither Appellants nor Appellants' legal representative know of appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the present appeal.

Status of Claims

Claims 1-24 are pending in this application. Claim 1, 8, 9, 16, and 23 are the pending independent claims at issue in this appeal.

This appeal is taken from the final rejection of claims 1-24. Appendix A presents the claims at issue in the appeal.

No claims are allowed.

Status of Amendments

No amendments after the final rejection have been submitted to, or entered by, the Examiner.

Summary of Claimed Subject Matter

The present application contains five independent claims. Independent claim 1 is directed a process for multiplexing of applications. The method of

claim 1 comprises a combination of steps including providing at least one access server that has access to at least one application, the at least one application capable of having a plurality of running instances, each of the instances capable of receiving and processing requests for a first service provided by the application during a session with a client. Such an access server is described, for example, at page 28 lines 6-12, and page 31, lines 18 - 29 and illustrated in Fig. 11. The method of claim 1 further includes the steps of: receiving a request from at least one client at the access server to access the first service provided by the at least one application; and based on the received request, establishing a communication link between the at least one access server and the at least one client, as illustrated in Fig. 11 and described at page 28, lines 29 - 32, and page 29, line 24 through page 30, line 7. The method of claim 1 further includes the step of storing the received request in an input request queue with other received requests, wherein the number of received requests may be greater than the number of running instances, as illustrated in Fig. 11 and described at page 29, lines 3-19. Claim 1 further includes the step of checking for an available communication path to the requested application, an available communication path being present when an instance of the requested application is available and ready to accept a new request, as described at page 31, lines 23 - 29 with reference to Fig. 11. When a communication path is available, a communication path is established between the input request queue and the at least one application, the stored request removed from the input request queue, the stored request send to the requested application; and a communication path established between the client and the requested application, thereby establishing a session with the client and the requested application and providing the first service to the client, as described at page 29, line 19, through page 31 line 2. Independent claims 16 and 23 are directed to a computer program products comprising a computer useable medium including code embodied therein for processing data in a manner similar to the process of claim 1.

Independent claim 8 is directed to a method performed on at least one processor for multiplexing applications. The method of claim 8 comprises a combination of steps including initializing at least one requests handler and at least one application handler, as described at page 47, line 31, through page 48 line 2, with reference to Fig. 18. The method of claim 8 also comprises accepting at least one request from at least one client to access a first service provided by a first application; passing the accepted request to an initialized request handler; completing a service request based on the passed accepted request; and putting the completed service request in an input queue associated with the first service, as described at page 48, lines 5 – 10 with reference to Fig. 18. Claim 8 continues with the steps of using an application handler to get the completed service request put in the input queue, the number of completed service requests in the input queue being greater than the number of applications capable of processing the completed service requests, as described at page 48, lines 9 – 13, with continued reference to Fig. 18. The service request is sent to the first application when the first application is available to process the service request; establishing a session with the first application; and establishing a communication link between the client and the first application, as described at page 48 lines 13 - 16, with reference to Fig. 18.

Independent claim 9 is directed to an apparatus for service multiplexing, the apparatus comprising: at least one access server capable of providing access to at least one application providing at least a first service that has an associated cost; the at least one access server comprising at least one agent and at least one service concentrator; and the at least one service concentrator comprising at least one application handler, at least one input service queue, and at least one request handler, such that the at least one access server is adapted to receive multiple requests from multiple clients to establish a session with the first service at the at least one application, as described at page 48, lines 2-21 with reference to Fig. 18. Claim 9 further requires that the at least one service concentrator is adapted

4

to multiplex the multiple requests to establish sessions with the first service at the at least one application and thereby divide the cost associated with the first service among the multiple clients, as described at page 41, lines 8 - 18..

Issues

The issue presented by the present appeal is:

Whether the Examiner erred in rejecting claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over United States Patent Number 6,477,561 to Robsman (hereinafter referred to as "Robsman") in view of United States Patent Application Publication Number 2001/0030970 to Wiryaman (hereinafter referred to as "Wiryaman").

Grouping of Claims

Appellant submits that all of the claims on appeal do not stand or fall together. The patentability of independent claims 1, 16, and 23 will be argued together, and the patentability of independent claims 8 and 9 will be argued together.

Argument

A. Summary of the Examiner's Final Rejection

The Examiner rejected claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable and obvious over Robsman in view of Wiryaman.

In particular, the Examiner rejected the claims, asserting that Robsman discloses a dispatch/pool manager that has access to an application that is capable of rinning a plurality of threads, with the dispatch/pool manager receiving and storing requests in an input request queue. The Examiner states that Robsman checks for an available thread, removes a stored request, and sends the stored

5

request to the available thread. The Examiner notes that <u>Robsman</u> does not specifically disclose establishing communications as claimed, and Examiner relies on <u>Wiryaman</u> for disclosing a link between a server and a client. According to the Examiner, it would have been obvious to combine a server as disclosed by <u>Wiryaman</u> into <u>Robsman</u> in order to provide a communication link between a client and server.

B. <u>Summary of the References Cited by the Examiner</u>

      <u>U.S. Patent No. 6,477,561 to Robsman</u>

      <u>Robsman</u> is directed to thread optimization on a computer capable of executing multiple execution threads. Particularly, Robsman is directed to varying the number of available threads that may be used for processes running on a computer, and describes a method of calling a "gating function" for each thread that is executed. As discussed in Robsman at col. 4, lines 57-59, "two function calls are inserted in the threads themselves to regulate the number of threads that are active at any given time." Robsman specifically describes, at col. 4 line 67 through col. 5 line 1 that the "functions keep a current count of the number of 'active' execution threads." Furthermore, Robsman describes in col. 5 lines 4-9 that "[t]he functions also maintain a variable limit of the number of active execution threads. Before allowing a thread to continue, the gating function compares the number of active threads to a variable limit. If the limit has already been met, the thread is temporarily delayed." Robsman goes on to describe, with respect to Fig. 5 the adjustment of the number of available threads. As described at col. 6 lines 11-39, the adjustment of the number of available threads is based on processor utilization at the time the adjustment function is called. In such a manner, the system of Robsman controls the number of active threads to provide efficient processor usage.

U.S. Patent Application Publication No. 2001/0030970 to Wiryaman

Wiryaman is directed to a network access device. The access device optimizes network traffic, or bandwidth, by examining the routing information (e.g. network addresses) on packets sent over the network and seeks to improve network performance by a combination of prioritization and proxying. As described at page 2, paragraph 49, "[a]ccess device 220 is a device that monitors traffic flowing through it, implements a user interface for setting configurable policies based on the characteristics of the monitored traffic, and enforces the configured policies." Further, paragraph 50 on page 3 explains "the policies that are enforced by access device 220 relate to allocation and use of communication resources related to communication passing between LAN 130 and WAN 110." The policies described include prioritization, where some packets are allowed to proceed over the network (WAN 110) while others are held back to reduce congestion, as described at page 4, paragraphs 63-64. Wiryaman discloses that a policy table may be used that specifies how different classes of inbound or outbound data flows are to be processed. The policies may also include proxying, where packets of data are re-routed to less busy/congested destinations that are deemed to be equivalent to the original destination recorded in the incoming data packet, as described at paragraph 65 on page 5. Importantly, Wiryaman is directed to data packets transmitted over a data network, in which traffic over one or more network devices is sought to be optimized.

C. Claims 1, 16, and 23 are patentable

Independent claim 1 is directed to a method for multipleximg applications. Independent claims 16 and 23 are directed to a computer program products comprising a computer useable medium including code embodied therein for processing data in a manner similar to the process of claim 1.

As is well established, in order to establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the

prior art. See MPEP §2143, citing In re Royka 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Furthermore, a determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the claimed invention. There must be a teaching or suggestion within the prior art, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources of information to select particular elements, and to combine them in the way that they are combined by the inventor. See Heidelberger Druckmaschinen AG v. Hantscho Commercial Prods., Inc., 21 F.3d 1068, 1072, 30 USPQ2d 1377, 1379 (Fed. Cir. 1994) ("When the patented invention is made by combining known components to achieve a new system, the prior art must provide a suggestion or motivation to make such a combination.").

Appellants submit that the Examiner has not established prima facie obviousness of claim 1. As explained above, Robsman is directed to thread optimization on a computer capable of executing multiple execution threads. Robsman optimizes processor usage through a gating function that is included in each thread. Robsman does not contemplate an access server that checks for communication paths to particular application and establishing a session between a client and application, because Robsman is directed to controlling threads to allocate processing resources. To the contrary, the present invention, as claimed in claim 1, is directed to multiplexing an application in which a client establishes a session with an application in order to receive a first service provided by the application. In such a manner, the first service is available for use by the requesting client through the access server. The access server, as required by the claim, stores received requests in an input request queue, checks for an available communication path to the application, establishes a communication link between the application and the requesting client when a communication path is available, thereby establishing a session with the first application and the client. In such a manner, multiple clients may access the first service that is provided by a

8

particular application. Importantly, the access server queues multiple requests for the first service and establishes a communication link to the application when a communication path is available to the application. The Examiner appears to interpret the claimed running instances of an application as being threads, and cites <u>Robsman</u> col. 4 lines 41-54 as teaching such threads. The Examiner goes on to state that <u>Robsman</u> checks for available threads and sends a stored request to the available thread. To the contrary, the present invention, as claimed in claim 1, requires establishing a *communication link* between the server and client, *checking for an available communication path* to a requested application and, when a communication path is available, establishing a *communication path* between the client and requested application thereby establishing a session. <u>Robsman</u> merely monitors the number of available threads, and adjusts the number of available threads based on processor utilization.

  <u>Wiryaman</u>, as discussed above, is directed to a network access device that optimizes network traffic by examining the routing information (e.g. network addresses) on packets sent over the network and seeks to improve network performance by a combination of prioritization and proxying. Importantly, the cited references contain no disclosure of establishing a communication link between a server and a client, checking for available communication paths to an application, and establishing a session with a client and requested application by establishing a communication path between the client and requested application in order to provide a service to the client. Thus, none of the cited references, taken alone or in combination, contain any teaching or suggestion for all of the steps of the method as claimed in claim 1.

  For at least the above-identified reasons, it is respectfully submitted that claim 1 is patentably distinct from the cited references either alone or in any reasonable combination thereof. Claims 2-7 all depend either directly or indirectly from claim 1 and, by virtue of dependency, are patentably distinct from the cited references either alone or in any reasonable combination thereof.

Claims 16 and 23 contain limitations similar to those described for claim 1, and it is submitted that these claims are also patentably distinct from the cited references. Claims 17-22, and 24 depend either directly or indirectly from claims 16 and 23, respectively, and by virtue of dependency are also patentably distinct from the cited references either alone or in any reasonable combination thereof.

D. Claims 8 and 9 are patentable

Independent claim 8 is directed to a method for multipleximg applications utilizing request and applications handlers. A request from a client is accepted and passed to the request handler, and a service request is completed based on the accepted request. The service request is placed in an input queue associated with the service, and the application handler is used to get the completed service request into the input queue. A session with the application is established when the application is available to process the service request, and a communication link is established between the client and application. Independent claim 9 is directed to an apparatus for service multiplexing comprising an agent and a service concentrator, the service concentrator comprising an application handler and a request handler.

As mentioned above, in order to establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. See MPEP §2143, citing In re Royka 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Furthermore, a determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the claimed invention. There must be a teaching or suggestion within the prior art, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources of information to select particular elements, and to combine then in the way that they are combined by the inventor. See Heidelberger Druckmaschinen AG v.

Hantscho Commercial Prods., Inc., 21 F.3d 1068, 1072, 30 USPQ2d 1377, 1379 (Fed. Cir. 1994) ("When the patented invention is made by combining known components to achieve a new system, the prior art must provide a suggestion or motivation to make such a combination.").

Appellants submit that the Examiner has not established prima facie obviousness of claims 8. As explained above, Robsman is directed to thread optimization on a computer capable of executing multiple execution threads. Robsman optimizes processor usage through a gating function that is included in each thread. Robsman does not contemplate request and application handlers as claimed. Robsman is directed to controlling threads to allocate processing resources. To the contrary, the present invention, as claimed in claim 8, is directed to multiplexing an application in which request and application handlers receive requests and establish a session with an application in order to receive a service provided by the application.

Wiryaman does not cure the deficiencies of Robsman. Wiryaman, as discussed above, is directed to a network access device that optimizes network traffic by examining the routing information, and seeks to improve network performance by a combination of prioritization and proxying. Importantly, the cited references contain no disclosure of a request handler and an application handler, as required by the claim. Thus, none of the cited references, taken alone or in combination, contain any teaching or suggestion for all of the steps of the method as claimed in claim 8.

For at least the above-identified reasons, it is respectfully submitted that claim 8 is patentably distinct from the cited references either alone or in any reasonable combination thereof.

Claim 9 contains application and request handler limitations similar to those described for claim 8, and it is submitted that claim 9 is also patentably distinct from the cited references. Furthermore, claim 9 requires that the service concentrator be adapted to multiplex the multiple requests and divide the cost

associated with the service among multiple clients, which is not taught or suggested by the cited references. Claims 10-15 all depend either directly or indirectly from claim 9 and, by virtue of dependency, are patentably distinct from the cited references either alone or in any reasonable combination thereof.

E. The Cited References Are Not Properly Combinable

      Furthermore, it is submitted that there is no suggestion or motivation to combine the references. As discussed above, Robsman is directed to thread optimization, and in particular to controlling the number of threads executing on a processor based on processor utilization. Wiryaman, as discussed above, is directed to a network access device that optimizes network traffic by examining the routing information on packets sent over the network. Neither of the references provide any suggestion or motivation for combining the references. Given that there is no suggestion or motivation in the references themselves, the Examiner must find that the suggestion or motivation to combine the references would be within the knowledge generally available to one of ordinary skill in the art. The Examiner states that the invention of Robsman is "directed to accessing server application over the network (by a client)." The Examiner then states that the combination of Robsman and Wiryaman would thus obvious to one of skill in the art. However, the Examiner is misstating the invention that Robsman is directed to. Robsman is directed to *thread optimization*, and not "accessing server application over the network" as suggested by the Examiner.

      Thus, it is submitted that, given the cited references are from vastly different fields of endeavor (thread optimization and routing of network traffic), one of ordinary skill in the art would have no suggestion or motivation to combine the references in the manner suggested by the Examiner. It appears that in the present case the only suggestion for the Examiner's combination of the cited references improperly stems from the Appellant's own disclosure and not from the cited references themselves. This is an inappropriate standard for

obviousness. A determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the claimed invention. There must be a teaching or suggestion within the prior art, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources of information to select particular elements, and to combine them in the way that they were combined by the inventor. See <u>Heidelberger Druckmaschinen AG v. Hantscho Commercial Prods., Inc.</u>, 21 F.3d 1068, 30 USPQ2d 1377 (Fed. Cir. 1994).

Therefore, it is respectfully submitted that claims 1-24 is patently distinct from the cited references either alone or in any reasonable combination thereof.

Request:

Reversal of the Examiner's final rejection of claims 1-24 is respectfully requested for the above-stated reasons.

Signed this 11th day of September 2006.

Respectfully submitted,

Kenneth C. Winterton, Reg. No. 48,040
HOLLAND & HART LLP
Post Office Box 8749
Denver, Colorado 80201
(303) 473-2700

Claims 1-24 involved in this Appeal read as follows:

1. A method performed on at least one processor for multiplexing applications, the method comprising the steps of:

providing at least one access server that has access to at least one application, the at least one application capable of having a plurality of running instances, each of the instances capable of receiving and processing requests for a first service provided by the application during a session with a client;

receiving a request from at least one client at the access server to access the first service provided by the at least one application;

based on the received request, establishing a communication link between the at least one access server and the at least one client;

storing the received request in an input request queue with other received requests, wherein the number of received requests may be greater than the number of running instances;

checking for an available communication path to the requested application, an available communication path being present when an instance of the requested application is available and ready to accept a new request;

when an available communication path is available, establishing the communication path between the input request queue and the at least one application;

removing the stored request;

sending the stored request to the requested application; and

establishing a communication path between the client and the requested application, thereby establishing a session with the client and the requested application and providing the first service to the client.

2. The method according to claim 1, further comprising the step of:

identifying a media transmission protocol based on the received request,

wherein the established communication link is capable of transmitting the identified media transmission protocol.

3. The method according to claim 2, further comprises the steps of:

verifying an accuracy of transmitted data; and

re-transmitting inaccurate data.

4. The method according to claim 1, wherein the establishing the communication link step uses,

at least one of session initiation protocols, H.323 protocols, MGCP protocols, MEGACO protocols, and H.248 protocols.

5. The method according to claim 2, wherein the identifying the media transmission protocol uses,

session description protocols.

6. The method according to claim 2, wherein the identified media is a real-time transport protocol.

7. The method according to claim 1, wherein the receiving the request step further comprises:

accepting a request at a request handler;

generating a service request; and

transmitting the generated service request to the input request queue for storage.

8. A method performed on at least one processor for multiplexing applications, the method comprising the steps of:

initializing at least one requests handler and at least one application handler;

accepting at least one request from at least one client to access a first service provided by a first application;

passing the accepted request to an initialized request handler;

completing a service request based on the passed accepted request;

putting the completed service request in an input queue associated with the first service;

using an application handler to get the completed service request put in the input queue, the number of completed service requests in the input queue being greater than the number of applications capable of processing the completed service requests;

sending the got completed service request to the first application when the first application is available to process the service request;

establishing a session with the first application; and

establishing a communication link between the client and the first application.

9. An apparatus for service multiplexing, the apparatus comprising:

at least one access server capable of providing access to at least one application providing at least a first service that has an associated cost;

the at least one access server comprising at least one agent and at least one service concentrator; and

the at least one service concentrator comprising at least one application handler, at least one input service queue, and at least one request handler,

such that the at least one access server is adapted to receive multiple requests from multiple clients to establish a session with the first service at the at least one application and the at least one service concentrator is adapted to multiplex the multiple

requests to establish sessions with the first service at the at least one application and thereby divide the cost associated with the first service among the multiple clients.

10. The apparatus according to claim 9, wherein the at least one agent comprises:
at least one SIP user agent.

11. The apparatus according to claim 10, wherein the at least one agent comprises:
at least one SDP agent.

12. The apparatus according to claim 11, wherein the at least one agent comprises:
at least one MTP agent.

13. The apparatus according to claim 12, wherein the at least one MTP agent comprises:
real-time transport protocols.

14. The apparatus according to claim 9, wherein the at least one service concentrator further comprises:
at least one service output queue.

15. The apparatus according to claim 9, further comprising:
at least one transmitting client to transmit a service request; and
at least one receiving client to receive a processed request.

16. A computer program product comprising:

a computer usable medium including computer readable code embodied therein for processing data to control at least one requests for access to at least one application, the computer usable medium comprising:

a request receiving module configured to receive at least one request for access to at least a first service provided by a first application;

a communication establishing module configured to establish a communication link with at least one client requesting access to the first application;

a storing module configured to store the at least one received request;

a checking module configured to check whether a communication path that is capable of allowing access to the first application is available; and

the communication establishing module further configured to establish a communication link with the first application, wherein the number of requests for access to the first application are capable of being greater than the number of requests capable of being processed by the first application.

17. The computer program product according to claim 16, further comprising:

a service concentration module configured comprise:

at least one request handler;

the at least one request handler generating at least one service request to be stored in the storing module; and

at least one application handler, such that the at least one application handler removes the stored request and transmits the stored request to the at least one application for processing.

18. The computer program product according to claim 16, wherein the communication module is further configured to output at least one processed request to at least one address indicated by the at least one client.

19. The computer program product according to claim 16, wherein the storing module is further configured to store at least one processed request prior to delivery.

20. The computer program product according to claim 17, further comprising:
a sip agent module configured to provide call control.

21. The computer program product according to claim 20, further comprising:
a sdp agent module configured to provide session descriptions,
such that the sip agent module directs the at least one request to a compatible request handler module.

22. The computer program product according to claim 21, further comprising:
a media transport protocol agent configured to provide transport protocols.

23. (previously presented): A computer program product comprising:
a computer usable medium including computer readable code embodied therein for processing data to control at least one requests for access to at least one application, the computer usable medium comprising:
a request receiving module configured to receive at least one request for access to a first service provided by the at least one application;
a first communication establishing module configured to establish a communication link with at least one client requesting access to the first service provided by the at least one application;
a storing module configured to store the at least one received request;
a checking module configured to check whether a communication path that is capable of allowing access to the at least one application; and

a second communication establishing module configured to establish a communication link with the at least one application and thereby establish a communication link between the client and the application, wherein the first communication establishing module is configured to establish more communication links than the second communication establishing module.

24. The computer program product according to claim 23 further comprising:

a third communication establishing module configured to establish a communication link with at least one address to receive at least one processed request.